Lecture Notes in Social Networks

Mehmet Kaya
Şuayip Birinci
Jalal Kawash
Reda Alhajj *Editors*

Putting Social Media and Networking Data in Practice for Education, Planning, Prediction and Recommendation

Springer

Lecture Notes in Social Networks

More information about this series at http://www.springer.com/series/8768

Mehmet Kaya • Şuayip Birinci • Jalal Kawash •
Reda Alhajj
Editors

Putting Social Media and Networking Data in Practice for Education, Planning, Prediction and Recommendation

Springer

Editors
Mehmet Kaya
Computer Engineering
Fırat University
Elazığ, Turkey

Şuayip Birinci
Ministry of Health
Ankara, Turkey

Jalal Kawash
Department of Computer Science
University of Calgary
Calgary, AB, Canada

Reda Alhajj
Department of Computer Science
University of Calgary
Calgary, AB, Canada

ISSN 2190-5428 ISSN 2190-5436 (electronic)
Lecture Notes in Social Networks
ISBN 978-3-030-33697-4 ISBN 978-3-030-33698-1 (eBook)
https://doi.org/10.1007/978-3-030-33698-1

This Springer imprint is published by the registered company Springer Nature Switzerland AG.
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

Social networks (SN) have brought an unprecedented revolution affecting how people interact and socialize. SN have invaded almost all domains: lifestyle, medicine, business, education, politics, activism, and more. The result is billions of SN users. For example, Twitter claimed to have 321 million monthly active users in 2018, and Facebook had 2.41 billion monthly active users as of June 2019. Online social media (OSM), media produced by SN users, have offered a real and viable alternative to conventional mainstream media. OSM are likely to provide "raw," unedited information and the details can be overwhelming with the potential of misinformation and disinformation. Despite these dangers, OSM are leading to the democratization of knowledge and information. OSM are allowing almost any citizen to become a journalist reporting on specific events of interest. This is resulting in unimaginable amounts of information being shared among huge numbers of OSM participants. For example, Facebook users are generating several billion "likes" and more than 100 million posted pictures in a single day. Twitter users are producing more than 6000 tweets per second. The size of the data generated presents increasing challenges to mine, analyze, utilize, and exploit such content. For example, analyzing the prohibitively long clickstreams in e-commerce applications and sifting through an overwhelming number of research articles become very challenging. At the same time, this explosion is opening doors to new exploitation and application of the generated data. For instance, various recommendation systems can make use of these data mines to generate more accurate and relevant recommendations.

This book includes twelve contributions that examine several topics related to the utilization and exploitation of SN and OSM. The topics are emergency evacuation scenarios, creating research groups, recommendation for food venues, clickstream analysis for e-commerce, event detection, fraud processes, scientific article recommendation, popular vs. unpopular activities, alternative navigation route recommendation, sentiment analysis, clickbait analysis, and the detection of anomalies.

Modeling crowd behavior in emergency evacuation scenarios is the subject of a chapter by Sahin and Alhajj. Using a belief-desire-intention model, they propose a multiagent system, which works with partially observable environments by the agents. Individuals perceive partial information about the environment and other evacuees continuously, and a belief set is created accordingly. A preconstructed set of plans is filtered using the current beliefs to find subtasks. Test scenarios and reported results are encouraging.

Green et al. report on partnering with students to perform a research project that investigated supporting students transitioning from first year to a Bachelor of Primary Education program. Students and their academic mentor were connected through a Facebook group, whose aim is to provide support to the group of transitioning students. Utilizing social media, students could access support from their peers and university staff and generate a community of learners.

Recommending feed venues using multilingual social media content is the subject of a chapter by Siriaraya et al. The system makes use of region-based popularity of venues, user rating, and tweet sentiment analysis in order to generate the recommendations. The system is experimentally validated using 26 million tweets from different European countries for four different recommendation approaches.

Xylogiannopoulos et al. embark on simplifying e-commerce analytics by discovering hidden knowledge in big data clickstreams. The massive number of the combination of online users, retailers, and products makes clickstream analysis very challenging. The authors address this issue by significantly modifying and upgrading a sequential frequent itemsets detection methodology. The result is a method that can do the analysis using very limited computational power, such as a desktop computer. They demonstrate the efficiency of their methods using 10 billion records related to top US online retailers.

Event detection in communities extracted from communication networks is the subject of a chapter by Aktunc et al. They focus on social interactions in communities to detect events; specifically they focus on tracking the change in community structure within temporal communication networks. Many versions of the community change detection methods are developed using different models. Empirical analysis shows that community change can be used as an indicator of events, and the ensemble model further improves the event detection performance.

Thovex looks at chasing undetected fraud processes. Employing deep probabilistic networks, hidden fraud activities can be detected. Inspired by waves temporal reversal in finite space, offline analysis and mining of big hidden data are tackled. Experiments are encouraging and the proposed model may introduce an alternative in artificial intelligence for a new generation of applications.

Betül Bulut et al. tackle the problem of recommending relevant scientific articles to researchers. Finding good matches is becoming more challenging with the increasing number of publications. An article recommendation system is presented in this chapter. This system takes into consideration the researcher's previous downloaded articles and the field of study. As a result of the study, the proposed method has achieved more successful results compared to other existing methods.

Patterns of behavior of individuals and groups in community-driven discussion platforms, such as Reddit, is the subject of a chapter by Thukral et al. Using statistical analysis, they provide interesting insights about a large number of posts that failed to attract other users' attention despite their author's behavior, they provide more insights about short-lived but highly active posts, and they analyze controversial posts. Conclusions include understanding how social media evolves with time.

Using Google maps history logs to mine habitual user choices is the subject of a chapter by Varlamis et al. GPS logs are analyzed for durations of stay in certain locations and most frequent routes taken by the user. Using a trajectory partitioning method, they identify the most frequent subtrajectories followed by the user. This in turn allows recommendation for alternative travel routes, improving the user experience.

The subject of sentiment analysis is covered in a chapter by Martin-Gutierrez et al. Due to the limits of standard sentiment analysis (such as dependence on the size and quality of the training set and prelabeling), the chapter proposes a new methodology to retrieve text samples from Twitter and automatically label them. Experimental analysis applied to various Twitter conversations yields promising results.

Geçkil et al. study "clickbaits," a mechanism to manipulate a reader to click on links that lead to unwanted or irrelevant content. They look at how such baits can be detected on news sites. In this study, the data was gathered from news websites of media organizations and Twitter to detect clickbaits. The method employed yields a high accuracy rate of detection.

Bouguessa takes on the problem of identifying and automatically detecting anomalous nodes in networks. Nodes are represented with feature vectors, representing neighborhood connectivity. A probabilistic framework that uses the Dirichlet distribution is employed to anomalies. The result is an automatic discrimination between normal and anomalous nodes, rather than providing a ranked list of nodes and delegating the detection to the user. Experiments on both synthesized and real network data illustrate the suitability of the proposed method.

To conclude this preface, we would like to thank the authors who submitted papers and the reviewers who provided detailed constructive reports which improved the quality of the papers. Various people from Springer deserve large credit for their help and support in all the issues related to publishing this book.

Elazığ, Turkey — Mehmet Kaya
Ankara, Turkey — Şuayip Birinci
Calgary, AB, Canada — Jalal Kawash
Calgary, AB, Canada — Reda Alhajj
September 2019

Contents

Contributors

Aman Agarwal IIIT Delhi, New Delhi, India

Toyokazu Akiyama Kyoto Sangyo Daigaku, Kyoto, Japan

Riza Aktunc METU, Computer Engineering Department, Ankara, Turkey

Reda Alhajj Department of Computer Science, University of Calgary, Calgary, AB, Canada

Department of Computer Engineering, Istanbul Medipol University, Istanbul, Turkey

R. M. Benito Grupo de Sistemas Complejos, Escuela Técnica Superior de Ingeniería Agronómica, Alimentaria y de Biosistemas, Universidad Politécnica de Madrid, Madrid, Spain

Mohamed Bouguessa Department of Computer Science, University of Quebec at Montreal, Montreal, QC, Canada

Grigoris Bouras Department of Informatics and Telematics, Harokopio University of Athens, Athens, Greece

Betül Bulut Department of Computer Engineering, Fırat University, Elazığ, Turkey

Arnab Chatterjee TCS Research, New Delhi, India

Lipika Dey TCS Research, New Delhi, India

Michelle J. Eady University of Wollongong, Wollongong, NSW, Australia

Ayşe Geçkil Department of Computer Engineering, Fırat University, Elazığ, Turkey

Corinne A. Green University of Wollongong, Wollongong, NSW, Australia

Esra Gündoğan Department of Computer Engineering, Fırat University, Elazığ, Turkey

Pinar Karagoz METU, Computer Engineering Department, Ankara, Turkey

Panagiotis Karampelas Department of Informatics and Computers, Hellenic Air Force Academy, Dekelia Air Base, Acharnes, Greece

Tushar Kataria IIIT Delhi, New Delhi, India

Yukiko Kawai Kyoto Sangyo Daigaku, Kyoto, Japan

Buket Kaya Department of Electronics and Automation, Fırat University, Elazığ, Turkey

Mehmet Kaya Department of Computer Engineering, Fırat University, Elazığ, Turkey

J. C. Losada Grupo de Sistemas Complejos, Escuela Técnica Superior de Ingeniería Agronómica, Alimentaria y de Biosistemas, Universidad Politécnica de Madrid, Madrid, Spain

S. Martin-Gutierrez Grupo de Sistemas Complejos, Escuela Técnica Superior de Ingeniería Agronómica, Alimentaria y de Biosistemas, Universidad Politécnica de Madrid, Madrid, Spain

Emily McMillan University of Wollongong, Wollongong, NSW, Australia

Hardik Meisheri TCS Research, New Delhi, India

Ahmet Anıl Müngen Department of Computer Engineering, Fırat University, Elazığ, Turkey

Lachlan Munn University of Wollongong, Wollongong, NSW, Australia

Yusuke Nakaoka Kyoto Sangyo Daigaku, Kyoto, Japan

Coşkun Şahin Department of Computer Science, University of Calgary, Calgary, AB, Canada

Christos Sardianos Department of Informatics and Telematics, Harokopio University of Athens, Athens, Greece

Panote Siriaraya Kyoto Institute of Technology, Kyoto, Japan

Caitlin Sole University of Wollongong, Wollongong, NSW, Australia

Christophe Thovex DATA2B, Cesson Sevigné, France

Laboratoire Franco-Mexicain d'Informatique et d'Automatique, UMI CNRS 3175 – CINVESTAV, Zacatenco, Mexico D.F., Mexico

Sachin Thukral TCS Research, New Delhi, India

Ismail Hakki Toroslu METU, Computer Engineering Department, Ankara, Turkey

Iraklis Varlamis Department of Informatics and Telematics, Harokopio University of Athens, Athens, Greece

Ishan Verma TCS Research, New Delhi, India

Yuanyuan Wang Yamaguchi Daigaku, Ube, Japan

Konstantinos F. Xylogiannopoulos Department of Computer Science, University of Calgary, Calgary, AB, Canada

Crowd Behavior Modeling in Emergency Evacuation Scenarios Using Belief-Desire-Intention Model

Coşkun Şahin and Reda Alhajj

1 Introduction

Realistic crowd behavior modeling has received considerable attention from the research community over the last two decades. Researchers applied different methods to imitate human actions for specific cases. Some of these works create cognitive models which focus on psychological aspects, while others consider physical forces as the main component of the whole process. There are works which considered the other evacuees as the major factor during the decision making process. Thus, they built the system based on the interactions among individuals and by considering group behavior models. These interactions are affected by various factors, including ethnicity, personal risk, credibility, social status, emotional status, personality, etc. [1, 2].

As it is hard to model crowd behavior by using a single source of elements, hybrid solutions with multiple components are becoming more popular and attractive to reflect more realistic scenarios. However, the set of factors affecting the decisions of an individual is huge. Especially during emergency situations, there are more aspects to take into consideration. For instance, panic, fear and chaos may prevent evacuees from acting reasonably and may lead to unforeseen scenarios. Thus, realistic behavior modeling in these cases is mostly different from strategic planning.

C. Şahin (✉)
Department of Computer Science, University of Calgary, Calgary, AB, Canada
e-mail: coskun.sahin1@ucalgary.ca

R. Alhajj
Department of Computer Science, University of Calgary, Calgary, AB, Canada

Department of Computer Engineering, Istanbul Medipol University, Istanbul, Turkey
e-mail: alhajj@ucalgary.ca

M. Kaya et al. (eds.), *Putting Social Media and Networking Data in Practice for Education, Planning, Prediction and Recommendation*, Lecture Notes in Social Networks, https://doi.org/10.1007/978-3-030-33698-1_1

Another important aspect in realistic crowd behavior modeling is incorporating fuzziness in the framework. In real-time situations, taking an action may not lead to the intended results. Also, there may be some accidents as a result of being careless or due to false assumptions. For instance, it is possible to see people colliding at corners or falling from the stairs after missing a step. However, an extremely detailed cognitive and physical model would fail in producing real-time output. Because considering every single detail for every individual in a continuous decision making process cannot be handled in a restricted amount of time with limited resources. As the number of factors affecting an individual is enormously big, some behavior without obvious reasons can be modeled as a part of randomness. For example, seeing an evacuee running in the opposite direction may make the others scared so that they unintentionally start to do the same thing. They may follow him/her even if they do not know the actual reason for running in the opposite direction. Similarly, misinterpretation of observations could result in taking faulty actions.

Partial observability is another crucial part of a realistic simulation model because, in real world scenarios, perceptions of the evacuees may not be clear or easy to comprehend. Every type of input, including smell, visual data, sound and physical contact is received independently. It is the job of the individual to evaluate and relate them with each other. By combining different inputs, a human-specific set of beliefs is created. There could be many irrelevant data, such as noise from other humans, which should be ignored. Moreover, it is possible to retrieve misleading data. For instance, hearing something incorrectly from another person or any type of misunderstanding is not an unusual event, especially in a crowded environment.

Similar to the cognitive process, perceptions coming to a simulation agent should be treated as independent data and the agent should be responsible for gathering them together to form something meaningful. Restricting perception range for vision/sound and making the agent find its own way using the data received during exploration could lead to a better crowd behavior model. However, it is important to consider the idea that each individual has limited resources and every single data cannot be processed in a detailed way. This phenomenon is known as bounded rationality [3].

As a result of the points explained above, creating a hybrid model containing the crucial aspects of real world planning is the key for realistic agent modeling. The model should represent processes such as perception, cognition, goal seeking, task scheduling, collaboration and communication [4].

The aim of the work described in this chapter is to create a partially observable environment and agent model as part of our existing multi-agent emergency evacuation simulation system. Among different approaches, we selected the Belief-Desire-Intention (BDI) framework which provides a high-level planning mechanism. Thus, it allows for the abstraction of low-level individual tasks, such as going from one location to another. The following section discusses the reasons of selecting BDI instead of some other approaches. Actually, the results reported in this chapter demonstrate the effectiveness and applicability of this approach for real-world scenarios.

The rest of this chapter is organized as follows. Section 2 gives background information and some example systems with their adapted approaches. Section 3 covers the developed emergency detection system and integration using the evacuation simulation system. In Sect. 4, we discuss the proposed BDI model and we describe the implementation details. Example simulations and result analysis of the system are provided in Sect. 5. Finally, Sect. 6 is conclusions and future work.

2 Background and Related Work

Agent-based architectures are commonly used in crowd behavior modeling. The main reason is the flexibility to represent each agent by its own attribute set with different values. This leads researchers to create heterogeneous environments where agents perform actions independently. Moreover, the architecture gives designers the flexibility to model the interaction among individuals and to simulate the group behavior. Thus, it is possible to use both microscopic and macroscopic approaches.

Dawson et al. [5] used an agent-based simulation approach to analyze the effects of flooding on people and provided a risk-based flood incident management (FIM) model. The responses of an agent are determined based on the hydrodynamic and the other environmental changes during a flooding incident. They prefer a probabilistic finite state machine to model states, actions and transitions. Some other works also used a multi-agent model with a feature set, e.g., [6–9].

There are various techniques adopted for modeling the decision making process of an individual. Partially-observable Markov decision processes (POMDP), the Belief-Desire-Intention (BDI) [10] framework, and cognitive architectures have been widely used for different purposes in the past. The BDI framework basically depends on the philosophical explanation of how humans make decisions. The details of performing actions are omitted and the main focus is on the process itself. Cognitive architectures aim to model the way human brain works as precisely as possible. Thus, the process depends on the results of neurological studies conducted on cognitive processes. Another difference between these two approaches is their abstraction level. Cognitive methods are low-level formulations of problems while the BDI model consists of mental steps of taking the action; it ignores the other parts. As the technical details are omitted, it is easier for non-computer science researchers to understand the steps of the BDI model.

POMDPs are used to model a learning problem in a structured way. Differently from a classical MDP structure, an agent in a POMDP environment cannot directly observe the current state. Thus, it uses a probabilistic approach on its observations. Various methods which have been used for solving MDPs, such as policy iterations and value iteration algorithms, have also been adapted for solving POMDPs, as well. The main objective in these approaches is to find the optimal solutions which lead to the maximum reward in the environment.

The POMDP framework is an effective way of representing some learning problems. The reward mechanism provides feedback to the agent during the process.

However, finding optimal POMDP policies is usually intractable. In addition, it is not easy to define multiple major tasks or some optional sub-tasks. On the other hand, the BDI model does not have a rewarding mechanism, but it is practical to introduce multiple tasks. Its plan-set is not adaptive, and it is not guaranteed to provide optimal solutions within a changing environment. Thus, both approaches have advantages and drawbacks. Accordingly, hybrid models may be more desirable for specific cases [11–13].

In addition to emergency evacuation, adaptive intelligent agents are needed in various types of problems, especially for crowd modeling. Video games, business process simulations and animal colony simulations are some of the applications of adaptive agents in dynamic environments. An agent-based BDI model is one of the most common choices for this purpose. In addition to early popular applications, such as Procedural Reasoning System (PRS) [14] and dMARS [15], the BDI approach has been adopted in some other works. It is not uncommon to see some hybrid solutions where the BDI framework cooperates or is combined with other techniques.

Zhao et al. [16] utilized the agent based BDI model for an automated manufacturing system. Their agent model is in the role of an operator who is responsible for detecting errors in an automated shop floor control system. The traditional BDI framework has been extended to include a deliberator, a planner and a decision executor. They also introduced the confidence index of a human which indicates how successful the agent has been with its latest actions. In case of poor performance, it reconsiders every intention before applying actions. The proposed model is adaptive and dynamically changes its behavior with the changing environment.

Lee et al. [17] used a similar modified BDI framework for mimicking realistic human behavior. In addition to BDI, they utilized the Bayesian Belief Network (BBN), the Decision Field Theory (DFT) and Probabilistic Depth First Search (PDFS) for planning and performing actions. The BBN model is employed to capture probabilistic relationships and historical data about the environment. The DFT model provides mathematical representation of the preferences of agents during the decision making process. It is used to calculate the preference values of each option. PDFS is the plan selection mechanism based on the preference values. It generates every option for the current situation and searches for a suitable one in a depth first search manner. The authors applied the proposed model for an evacuation scenario of a terrorist bomb attack.

Okaya et al. [2] also integrated the BDI model with a different method. They combined Helbing's social force model [18] with an agent-based BDI structure to simulate physical forces, psychological states, environment knowledge of agents and inter-agent communication at the same time. The agents are modeled to receive visual and auditory input from the environment. Perceptions are used to generate human relation factors created as a union of personal risks, family context and adaptive plans. Civilians are created as part of one of three groups, namely an adult, a parent or a child. Depending on their types, their attitude towards the others and their collaborative abilities are formed. Social forces on an agent are calculated using Helbing's model and they are modified by the intentions of an agent and its attribute values.

Liu et al. [19] used navigational agents to group individuals in an evacuation area. Their model contains a knowledge-base with different sets of information. They created a two-layer control mechanism with a social force model and a cultural algorithm (CA) [20], which has been designed for modeling the cultural evolution process. The approach is effective in terms of dividing the crowd behavior simulation problem into sub-groups each of which is guided by a leader.

Adam et al. [21] studied Melbourne bush fire in 2009 by building two models based on a Finite State Machine (FSM) and the BDI framework. By addressing the same problem with different architectures, they compared the models in terms of their effectiveness. They argued the following, while FSM requires generating every possible state and the transitions between each other, the BDI model is more convenient for designing reactive agents. However, the BDI model is a more complex structure and may require greater computational power. Their comparison is based on multiple factors, including the statistics of the real incident and the simulated ones.

For some public events such as concerts, sports competitions and public places such as airports, majority of the individuals may be unfamiliar with the building. However, in an office where people see each other daily, relations among people and the fact that people know the environment cannot be ignored during egress. Many of the recent works rely on this fact, including the work of Valette et al. [22]. They created a heterogeneous environment of agents with social skills and emotions. Decision making is performed by the BDI model which also considers the attributes of an agent during the process. The proposed model has been applied to multiple real-world scenarios in order to evaluate its reliability.

3 Multi-Agent Emergency Evacuation Simulation Model

Our emergency evacuation simulation system is an agent-based model which is intended to be used for running egress simulations. The main objective is creating evacuation test case scenarios which are dangerous or impractical to build in real life. It is crucial to analyze every aspect of a building for emergency management purposes, even before starting the construction. In addition, this may be helpful for seating arrangements and exploring capacity limitations of buildings in case of emergencies.

The system uses a set of physical, mental and social attributes to model different types of humans in an evacuation process. Our primary concern in this part of the evacuation planning project is to model individuals and groups as realistic as possible. This way, different scenarios can be run and analyzed correctly. We mainly focus on crowd modeling and human decision making process. We leave the low-level aspects of physical interactions to the Unity game engine.

We use position, health, mobility, gender, age, stress level, panic level and social role attributes for each agent. Social role determines if the agent is a leader for the group or a follower. Depending on the physical abilities, an agent can choose resting, moving to a specific position or following a group/leader. Speeds of agents are calculated by a fuzzy logic engine which uses physical and mental features in fuzzy inference rules.

The model validation is done by conducting some experiments and observing particular individual and group behavior, such as selfish and collaborative attitudes. In addition, common phenomena like herding, arching and clogging are also detected. The model has been tested on a building floor with different numbers of evacuees while varying exit gate widths.

This architecture lacks of partial observability and agent-to-agent communication capabilities. In real world, especially during panic situations, it is not easy to interpret every input data correctly and in a fast way. Moreover, it is not a good idea for each agent to assume full knowledge about the building plan. Thus, in this work, our aim is to introduce a multi-agent BDI framework, which models individual planning and agent interactions in a more realistic way.

4 BDI Framework in Emergency Evacuation

4.1 BDI Framework

BDI framework is a common approach for building multi-agent systems where a human behavior model is necessary. It adopts the human decision making psychology by using beliefs, desires and intentions. Beliefs are the combination observational data with the information retrieved from other agents. Desires are the main goals of an agent. They tend to stay the same during the process, unless something unexpected occurs. Extraordinary cases could result in changes int the objective of an agent. In an emergency evacuation, for instance, the desire is leaving the building. However, a mother would mostly prefer saving her children, an action overrides the current desire. Similarly, for an agent with partial knowledge, desire may be expressed as reaching to a particular exit. However, if the pathway is blocked by obstacles or other evacuees, it may need to define a new desire. Intentions are simple actions or small checkpoints to achieve the final goals. For example, in order to leave a building, the first step for an agent could be exiting from the current room. Thus, it needs a partial plan to reach to the room door while staying away from obstacles and other evacuees. Typically, a BDI framework contains this type of predefined sub-plans which can be adopted by agents.

The need for a BDI framework arises because of the issues related to simple environment models, ignoring some crucial factors which exist in real world. First of all, in real world cases, the environment is highly dynamic. Possible presence of fire, smoke or flood and the other individuals performing actions should be considered continuously. A plan which is built based on the current snapshot of the environment could be obsolete while performing related actions. For example, in case of clogging in one of the doors, agent will not be able to continue. Thus, it should adapt its behavior according to the changes. Secondly, the amount of resources an individual can use is limited. Therefore, it may need to ignore some past information or irrelevant up-to-date data while deciding on next action. In addition, some perceptions could be noisy. This is the reason for calling any input as *beliefs* instead of *facts*. Thus, relying some observations for a long time could be harmful in case they are somehow misinterpreted.

4.2 Proposed Model

This part of the project aims to integrate a custom BDI framework to our previous work described in Sect. 3. The main reason is building a partially observable multi-agent model capable of adapting agents' behavior according to their perception. This type of approach is effective mainly when the state of the environment constantly changes. In addition, it will allow us to imitate psychological steps of processing input, using it for decision making and performing actions. The BDI architecture works with the existing modules in every part of the process. The proposed model is depicted in Fig. 1.

The architecture consists of a set of components which process, filter and prepare data for the other components. The main source of data is the environment itself. Each agent constantly collects information as *perception*. The major components of the system are:

(1) Input Manager Each agent continuously collects data from the environment via its sensors. Even though it is possible to process any kind of data in the following steps, the current setup only considers visual data. Ideally, every data type should include its propagation dynamics. For instance, in order to simulate smelling smoke, feeling fire, hearing the sound of other agents and the motion of flooding, additional dynamics should be defined. For visual data, the technique we use is sending multiple rays to different directions in the vision range of the agent and detecting the first object that the corresponding ray intersects with. This way, agents detect every obstacle, evacuee and gate seen in its specific vision range. Figure 2 shows two agents collecting visual data in a test environment. These objects are stored in the *input queue*. In addition to direct object detection, agents are capable of detecting components of the building and their connections. This provides locational awareness to agents. For instance, the bottom agent in Fig. 2 can detect that it is in *Room 2* and there are two close-by doors, namely *Door 1* and *Door2*. Initially, it does not know anything about the presence of neither the other agent nor the other building components. Thus input manager provides the information of what is around and where the agent in terms of belief. The filtered perceptions are passed to the *beliefs* list.

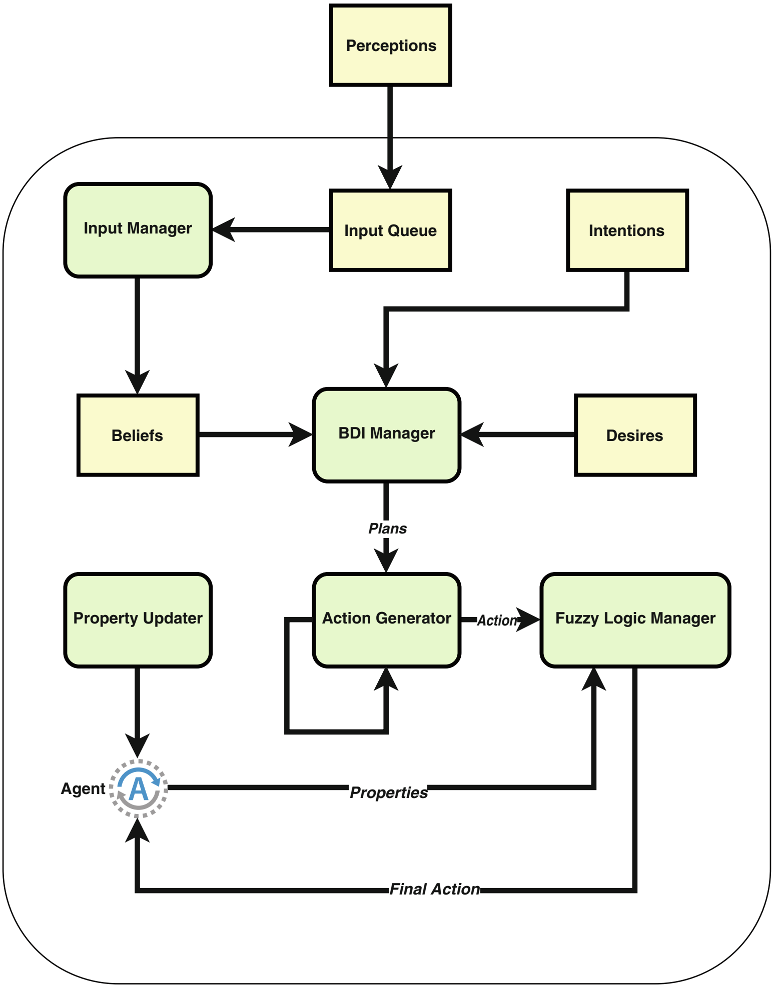

Fig. 1 Integrated BDI architecture

(2) BDI Manager BDI Manager uses the belief, desire and intentions sets to generate a plan for the corresponding agent. The belief set is populated by the *Input Manager* and contains environmental data including the current location of the agent, objects and dangers around. By default, the desire of an agent is leaving the building. However, it is possible to overwrite it. For instance, an agent may give up after an unsuccessful exploration and decide to follow a particular person. Intentions contain pre-constructed abstract plans to achieve the desires. They are defined in a high level manner. For instance, when the agent detected an unblocked exit gate, it should directly go there. On the other hand, if there is a door to another room instead of an exit gate, it could navigate through the door. By using the belief

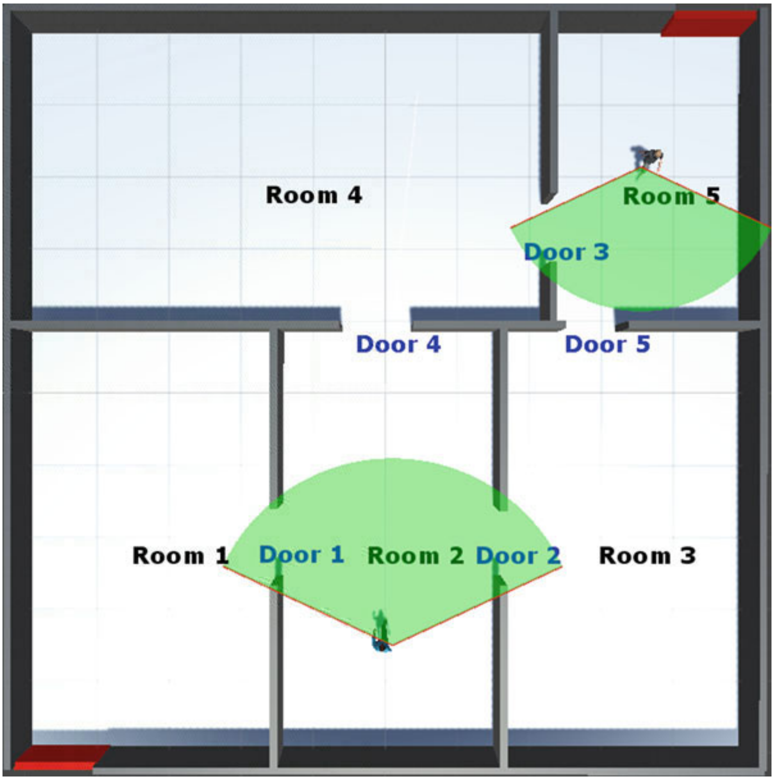

Fig. 2 Views of different evacuees

set, BDI Manager chooses the most appropriate intention that may lead to achieve the goals under the current circumstances. For this purpose, it also creates a partial map of the building using the observations. Figure 3 depicts a sample map of the bottom agent in Fig. 2 after some exploration. It shows that the agent was able to detect doors 1, 2, 4 and rooms 1, 2, 3. However, it does not know what is there after *Door 4*. Thus, it is an exploration opportunity for the agent. It may also decide to explore *Room 3* or even *Room 1* until it makes sure that there is no exit or other door(s) in this room.

(3) Action Generator *Action Generator* creates a set of appropriate actions for the current plan. For navigational actions, it uses the maps produced by the agent during exploration. For instance, if the agent decides to leave the room, action generator will check if a door is observed. If not, it will either decide to do random walk around to find an exit or follow another agent that is believed to be credible. If the agent believes that it is in danger, depending on the plans defined in the system, it will try to escape from the current location as soon as possible. Again, related actions are generated by this module, until the *BDI Manager* decides something else. Thus, this

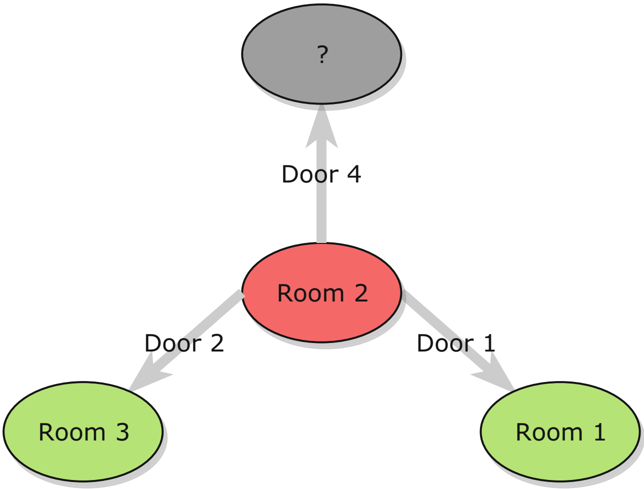

Fig. 3 Partial map example of the bottom agent after some exploration

module continuously generates actions until the current plan is achieved or changed. In case the BDI Manager updates or cancels the plan, the Action Generator starts from the beginning.

(4) Property Updater This module updates the attributes of the agent directly, depending on the current situation. Failure of attempting some actions could increase the panic level and running for some time may decrease the health level. On the other hand, any improvement in the state of the agent keeps it calm. In some cases, it may also directly affect the plan. For instance, if the agent is extremely tired, it has to take a rest in a safe place to be able to get out of the dangerous area without being hurt.

(5) Fuzzy Logic Manager This module is responsible for calculating the speed of the agent using fuzzy logic. It uses attributes of the agent in a fuzzy inference rule set after fuzzification. The resulting labels are defuzzified and crisp values are calculated.

The resulting system is highly adaptive and modular. It divides the evacuation problem into sub-problems and tries to solve them separately. It is possible to change even the major objectives. The system may continue operating under the current circumstances. Depending on the belief set, the *BDI Manager* creates a set of plans and passes it to the *Action Generator*. The action generator continuously feeds the fuzzy logic manager with actions. The *Fuzzy Logic Manager* finds the speed of the agent by using the current attribute values of the corresponding agent. It passes the final action to the agent again. The action generator generates the actions according to the current set of plans. If the BDI manager changes the plans, the action generator resets itself and starts working using the new set. This is the way to model the adaptive behavior of agents.

5 Experiments

In order to test reliability and effectiveness of our approach, we have conducted multiple experiments in a building floor with dimensions $100\,\text{m} \times 100\,\text{m}$. As depicted in Fig. 2, the floor contains five rooms and two exit gates. Eight different human models are used to make it easier to track group movements and specific individuals. Every model has its own default physical and mental attribute value ranges. The real values are assigned during initialization of the simulation. However, these values are kept the same for different experiment sets in order to eliminate having different observations as a result of attribute values. There are 140 individuals in the building. The primary objective is observing the behavior of evacuees under different configurations.

There are three different approaches we tested with multi-agent BDI framework with the fuzzy logic inference model:

- **Agents without any experience:** This setup aims to observe the behavior of agents when they have no prior knowledge about the environment. In addition, there is no cooperation among individuals. Thus, they need to find the rooms, doors and exit gates via exploration. These agents still use the BDI model, but their plan set does not contain anything related to the interaction with others.
- **Agents with partial environment knowledge:** This set of experiments aims to simulate the evacuation of individuals who have partial knowledge about the building plan. The idea is, even if it is his first time in the building, a person at least may roughly know the pathway to the gate he used to enter the building. Therefore, we assume that every individual has a plan, which is possibly not optimal, but as the crowd is not organized, it will possibly lead to some chaos.
- **Cooperative agents:** This case is for a scenario where there are experienced leaders who are knowledgeable about the evacuation and they organize the groups. These navigational agents represent educated emergency personnel. Thus, instead of moving independently, evacuees follow these leaders to egress gates. At the beginning, each evacuee starts searching for either an exit or a leader. Here it is worth mentioning that, it is possible for an agent not to find a leader and explore the area by itself as in the first test case scenario.

Each test case scenario has been simulated ten times and the number of evacuees who leave the building depending on the time graph is depicted in Fig. 4.

The results show that, in an organized set of groups that follows a trained leader, it is much faster to reach to an egress gate. They also decrease the possibility of common phenomena during evacuations, such as arching and clogging. While most of the crowd leaves the building less than 250 s, it takes more than 350 s both for exploring evacuees and partially experienced evacuees. In each set of experiments, there is a certain number of agent with low mobility. This explains the slow rate of evacuation for the latest individuals.

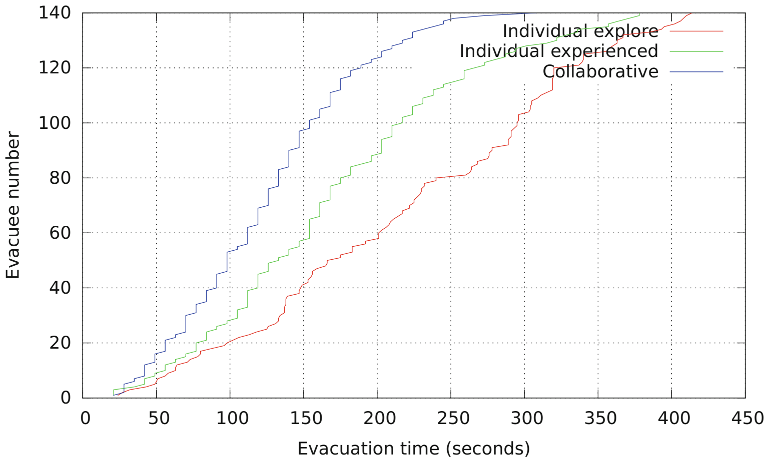

Fig. 4 Number of evacuees/evacuation time graphs for different approaches

Fig. 5 Clogging example during agent exploration

Figures 5 and 6 show snapshots of exploration agents without prior knowledge and cooperative agents respectively. Comparison between the flow in the doors for two cases clearly shows the effectiveness of organizing the groups. While explorer agents lead to clogging in the doors, the flow in the organized groups is much faster.